(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平10－254840

(43)公開日　平成10年(1998)９月25日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	15/16	４３０	Ｇ０６Ｆ	15/16	４３０Ｚ
	9/06	５５０		9/06	５５０Ｚ
	13/00	３５５		13/00	３５５
	15/00	３３０		15/00	３３０Ａ

審査請求　未請求　請求項の数７　ＯＬ　（全 11 頁）

(21)出願番号　特願平9－58651

(22)出願日　平成９年(1997)３月13日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　川村　旭
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

(74)代理人　弁理士　田中　治幸　（外２名）

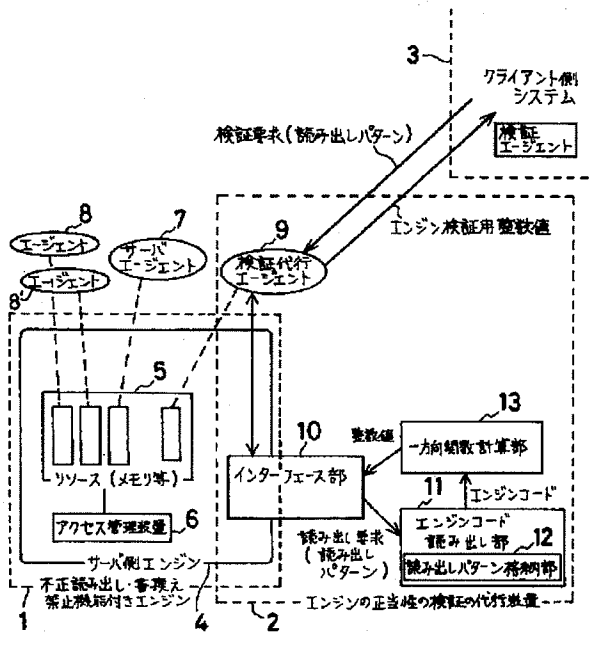
(54)【発明の名称】　モーバイルコード実行システム

(57)【要約】

【課題】　分散して存在する多数の計算機がネットワークで接続された環境における情報処理のセキュリティ技術に関し、特にモーバイルコードによるマルチエージェント方式での情報処理において、モーバイルコードの実行時に不正処理を行なわないように保証するモーバイルコード実行システムを提供することにある。

【解決手段】　モーバイルコードの実行に際して、モーバイルコード中の必要のない情報の読み出し及び書き換えを禁止する機構を持つ実行エンジンと、そのエンジンのコードを一方向関数に入力して得られる整数値を調べることで、エンジンコードが書き換えられていないことを検証する装置を備えるようにモーバイルコード実行システムを構成した。

本発明の原理的構成



1

【特許請求の範囲】

【請求項１】　モーバイルコードの実行に際して、モーバイルコード中の必要のない情報の読み出し及び書き換えを禁止する機構を持つ実行エンジンと、そのエンジンのコードを一方向関数に入力して得られる整数値を調べることで、エンジンコードが書き換えられていないことを検証する装置を備えることを特徴とするモードバイルコード実行システム。

【請求項２】　請求項１に記載のモーバイルコード実行システムにおいて、エンジンコード読み出し部に、読み出しパターン格納部を追加し、エンジンコードのすべてを読み出すのではなく、読み出しパターンで指定された番地のデータだけを読み出すことを特徴とするモーバイルコード実行システム。

【請求項３】　請求項１に記載のモーバイルコード実行システムにおいて、クライアント側のエンジン上で、サーバ側のエンジンの正当性を検証するエージェントを動作させ、利用するサーバが動作するエンジンに向けてモーバイルコードを送信する前に、その相手先のエンジンの正当性を検証することを特徴とするモーバイルコード実行システム。

【請求項４】　請求項３に記載のモーバイルコード実行システムにおいて、サーバ側のエンジン検証のためのメッセージ交換時に、相互の検証と、メッセージの電子署名と暗号化を行なうことを特徴とするモーバイルコード実行システム。

【請求項５】　請求項１に記載のモーバイルコード実行システムにおいて、送信するモーバイルコードの構成内容について、実行コードとして、エンジンの正当性の検証と検証後の暗号化されたコード等の復号化と復号化されたコードの実行を行うサーバ側のエンジン検証用エージェントのコードが格納され、持参したデータとして、暗号化された本来の実行コードとそのデータとが格納されており、その暗号化はサーバ側のエンジンコードを一方向関数に入力して得られる整数値を鍵として復号化できるように施されていることを特徴とするモーバイルコード実行システム。

【請求項６】　請求項５に記載のモーバイルコード実行システムにおいて、本来の実行コードとそのデータとの暗号化について、サーバ毎に固有の鍵を暗号化のときに付加し、暗号化に公開鍵を用いる場合には、サーバから得たデータを新たに持参データにするときに、クライアントまたは途中でそのデータを利用する必要のあるサーバの公開鍵暗号を用いた暗号化を行うことを特徴とするモーバイルコード実行システム。

【請求項７】　請求項１に記載のモーバイルコード実行システムにおいて、エンジンの自己検証装置を付加し、サーバのエンジンの起動時及び、稼動中は定期的に、エンジン外のエンジンコード読み出し部と一方向関数計算部でエンジンコードに依存した整数値を得て、正しい値

2

との比較を行なうことで、エンジンコードの書き換えの自己検証を行なうことを特徴とするモーバイルコード実行システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、分散して存在する多数の計算機がネットワークで接続された環境における情報処理のセキュリティ技術に関し、特にモーバイルコードによるマルチエージェント方式での情報処理において、モーバイルコードの実行時にそのモーバイルコードに対して不正処理を行わないように保証するモーバイルコード実行システムに関する。

【０００２】パーソナルコンピュータの発達及びコンピュータネットワーク技術の発達によって、大型計算機をメインホストとし、それに端末がぶら下がって、すべての情報処理をメインホストに依存する集中処理方式から、個々のパソコン・ワークステーションをフロントエンドとして、ネットワーク上に設置された複数のスーパーコンピュータを用いた計算サーバ・大規模な記憶装置を持つ計算機を用いたデータベースサーバを適宜選択して目的の情報処理を行うというクライアントーサーバ方式が主流となってきている。

【０００３】さらに、全世界的なコンピュータネットワークの技術動向として、オープンな規格によって、用いている計算機や通信媒体の枠を乗り越え、個々の計算機ネットワークの接続によって形成されたネットワークのネットワークであるインターネットが急速に拡大している。その動きは、規模の面だけでなく、利用の面からも専門家から一般の人々へと拡大し、インターネットは企業や国といった既存の枠組みを越え、全世界的な広がりを持ったコンピュータネットワークとして成長し続けている。また、インターネットの柔軟性と利便性を引継ぎながら、利用範囲を特定の組織内に限定したイントラネットも、昨今、普及してきている。

【０００４】そのようなインターネット・イントラネットの規模と利用分野両面における拡大につれて、クライアントーサーバ間の関係も、限られた範囲の閉鎖的・固定的なものから、ネットワーク上に存在する全てのものへと、時に応じて自由に接続・利用することができるものへ進化することが望まれるようになった。そのため、利用可能なプラットフォームに束縛されないようなインターフェースが必要となった。また、利用するユーザが、限られた専門家から一般の人達へと広がるにつれて、簡易なインターフェースで複数のネットワークの異なる種類のサーバを統合して扱うような高度なサービスを提供することが必要とされるようになった。それらの要望をみたすために、アクセスする側のエージェントが、各々のサーバを管理するエージェントとのメッセージの交換によって、高度な情報処理を実現するマルチエージェント方式が注目されている。

【０００５】また、携帯端末の普及によって、常にネットワークに接続しているわけではない場合への対応や、サーバ側の出来合いのエージェントとのメッセージ交換ではなく、こちらからエージェントを送り込んで高度で柔軟な情報処理を行うために、エージェント自体がネットワーク上を移動するというモーバイルコード（モーバイルエージェント）方式がマルチエージェント方式の一部として重視されている。

【０００６】

【従来の技術】マルチエージェントのシステムは、次のようにインプリメントされる。すなわち、各計算機上に少なくとも一つ、エージェントの生成・消滅やエージェント間の通信を司るエンジン（シェルやインタプリタ等）が置かれる。そして、各エージェントは、互いに独立的に機能し、かつ相互にメッセージの交換を行うように、プロセスやスレッド等の実体が与えられる。各エージェントの実体は、エンジンのプロセス内に存在する場合と、エンジンとは独立なプロセスである場合とがある。ネットワーク上の別の計算機上に存在するエージェントとのメッセージ交換は、ＴＣＰ（伝送制御プロトコル）またはＵＤＰ(User Datagram Protocol)等を用いて、エンジン経由で間接的に、またはエンジンを仲介としてエージェント間で直接行われる。

【０００７】モーバイルコード方式では、一方の計算機から、実行すべきエージェントのコードをメッセージとして送信し、受信した計算機で、そのコードを実行する。そのために、一般的には、送付されたコードを計算機のシステム上で直接実行するインプリメントはとられず、エンジンを、そのコードを実行するためのインタプリタ、またはバイトコードを実行するための仮想マシンとするインプリメントがとられる。そうするのは、マルチプラットフォームとするための抽象化と、後で述べる計算機のシステムのセキュリティ確保のためである。

【０００８】ネットワーク上に計算機が分散している環境、特に全世界的な広がりを持つネットワークであるインターネットに於ては、その柔軟性と利便性の裏返しとして、誰が、何時、何処から、如何なる手段で、何にアクセスして来て、何を為し、その結果の影響がどう出るのかについての可能性（危険性）は、無限にある。そのため、悪意または不注意によるアクセスを制限する必要がある。その実現のために、ネットワークのレベルでは、きまった範囲のネットワークの入口に、アクセスコントロール用計算機を用いたファイヤウォールを設置して、通信する相手先のネットワークや、通信プロトコルの種類に制限を加えることが行われている。

【０００９】また、マルチエージェントのエンジンのレベルでは、公開範囲の制限や、過剰占有によるシステムの動作不能防止のため、アクセスできるリソースの範囲・量・占有時間に制限を加えることや、コンピュータウイルス等のシステムへの不正アクセスを防止するために、計算機のシステムへは常に間接アクセスすることにしたり、モーバイルコードの実行前にコード検証を行うこと等が行われている。

【００１０】また、インターネットはその仕組み上、一般的にどのような経路を通るかが判らず、経路上での交信データの盗聴・改ざんや、悪意の第三者が本来の相手に成り済ます可能性・危険性がある。それを防止するために、交信内容の暗号化や、暗号技術を応用した相手の認証・改ざんされてないことを保証するための電子署名等の技術が用いられている。具体的には、秘密鍵暗号を用いるケルベロス、あるいは公開鍵・秘密鍵暗号を用いたＰＥＭ(Privacy-Enhanced Mail）やＰＧＰ(Pretty Good Privacy）等が存在する。

【００１１】

【発明が解決しようとする課題】モーバイルコードそれ自体が、高機能化するにしたがい、知的財産権確保の対象としての保護が必要になっていく。また、モーバイルコードはデータを伴った形でネットワークに発信され、ネットワーク上の複数のサービスサーバとメッセージ交換をしていくことによって、データの引き出し・処理等を次々に行っていく使い方も行われ得る。その場合、ネットワーク上のサービスサーバが常に信用できるとは限らないので、データ処理に必要ないデータの読み出しを防ぐ必要があり、また、あるサービスサーバから得たデータを他のサービスサーバには秘匿する必要も有り得る。

【００１２】しかし、従来のセキュリティ技術は、外部の不正アクセス等から、ネットワーク内のデータやシステムを守ることを目的としており、送信するモーバイルコードに関する保護は十分ではない。サービスを提供するサーバをエージェントの一つとしてしまい、サーバからモーバイルコードの内容を不正に覗き見できないようにしたエンジンもあったが、そのエンジンが本当にその制限機能を持っているかどうかを検証する手段は存在しなかった。

【００１３】

【課題を解決するための手段】本発明のモーバイルコード実行システムでは、サーバ側のシステムは、モーバイルコードを実行する際に、モーバイルコード中の実行コード部分および持参データ部分の不正な読み出し・書換えを禁止する機能付きのエンジンと、そのエンジンが正当なものであることの検証を代行する装置とから構成される。モーバイルコード中のデータの不正な読み出し書換えを禁止するエンジンの機能が正当であるかどうかの検証は、そのエンジンのコードをチェックすることにより行う。

【００１４】エンジンは、各エージェントが明示的に許可していない領域に他のエージェント等がアクセスできないように、各エージェントが実行中に使用するメモリなどのリソースのアクセス管理を行う管理装置を内部に

持つ。

【００１５】図１は、本発明の原理構成図である。図１において、１は本発明による不正読み出し・書換え禁止機能付きエンジン、２は本発明によるエンジンの正当性の検証の代行装置である。３はサービスを要求するクライアント側システムである。４は不正読み出し・書換え禁止機能付きエンジン１が実現されるサーバ側エンジンである。５はサーバ側エンジン４内に備えられているメモリ等のリソースである。６はリソース５へのアクセスを管理するアクセス管理装置、７ないし８はリソース５を用いて動作しているエージェントで、７はサーバエージェント、８，８′は他のエージェントである。９ないし１３はエンジンの正当性の検証の代行装置２を構成し、９はサーバ側エンジン４上で動作する検証代行エージェントである。１０はエンジン内のインタフェース部である。１１はエンジン外の一次記憶または二次記憶上に存在するエンジンコード読み出し部で、１２はエンジンコードから読み出すデータの番地を指定するパターンが格納される読み出しパターン格納部、１３はエンジンコード読み出し部１１から読み出したエンジンコードのデータを入力として関数計算し、固定長の整数値を出力する一方向関数計算部である。

【００１６】クライアント側システム３には以下の２通りの構成がある。その一つは、サーバ側のエンジンの正当性の検証を行うエージェントが、クライアント側のエンジン上で動作するものである。他の一つは、モーバイルコード自体の中に、暗号化された本来の実行コードと持参データとが有り、さらにエンジンの正当性の検証や、検証後の暗号化されたコード等の復号化と、そのコードの実行を行うためのエージェントのコードが有るというものである。そのいずれの場合においても、クライアント側には、サーバ側のエンジンのコードの情報が格納されている。

【００１７】モーバイルコードシステムは、利用するサーバが動作している計算機に、不正な読み出し・書換えから保護すべき実行コード・持参データを送信する。したがって、モーバイルコードのセキュリティの確保は、モーバイルコードが実行されるエンジンに於て行われなければならない。しかし、一般的には、サーバ側エンジン４は、クライアント側システム３の管理下にはない。したがって、本発明では、サーバ側エンジン４として、不正な読み出し・書換えを不可能とするものを用い、クライアント側からは、サーバ側のエンジンが正当なものであるかを検証可能とすることで、クライアント側から送出するモーバイルコードのセキュリティを確保する。

【００１８】サーバ側エンジン４は、各エージェントが明示的に許可していない領域に他のエージェント等がアクセスできないように、各エージェントが実行中に使用するメモリなどのリソース５のアクセス管理を行うアクセス管理装置６を内部に持つ。かつ、サービスを提供するサーバは、サービスをエージェントの一つとしてインプリメントする。そうすると、サーバの各サービスから、モーバイルコードが明示的に指定したデータ以外の読み書きをするのが不可能となる。

【００１９】サーバ側エンジン４の上には、エンジン４の正当性の検証の代行を行う検証代行エージェント９が動作している。検証代行エージェント９は、クライアント側の検証エージェントの要求メッセージにしたがって、エンジン４内のインタフェース部１０を介して、エンジン外のエンジンコード読み出し部１１と一方向関数計算部１３でエンジンコードに依存した整数値を得て、その整数値をクライアントの検証エージェントに返す。ここで、一方向関数は、セキュアハッシュ関数・Messge Digest 関数とも呼ばれ、具体的には、ＭＤ４，ＭＤ５，ＳＨＡ，チェックサムＣＲＣなどである。一方向関数は、以下の性質を持っている。

【００２０】１．任意のバイト列を入力として固定長の整数値を出力
２．同じ出力を与える入力の発見が困難
３．逆関数を定義するのが困難

エンジンコードを不正処理実現のために書き換えると、一方向関数が出力する整数値が、元のエンジンコードのときと異なってしまうので、書換えを検出することができる。当然、偶然に同じ値になることは有り得るが、整数値の大きさが十分ならば、実用上問題のない確率に収められる。また、同じ出力を与える入力の発見が困難であるので、書換えたエンジンコードに、書換えの検出を逃れるためのつじつま合わせのための追加書換えを行うことは困難である。ここで、サーバ側のセキュリティの観点からみると、サーバ側がクライアント側に提供するのは、そのままでは限りなく無意味な整数値に過ぎない。クライアント側がエンジンコードの内容を知っていてはじめて、その整数値が意味を持つ。つまり、サーバ側にとっては、最小限の情報の提供で、検証が可能ということになる。

【００２１】クライアント側の検証エージェントは、そのエンジンコードに対応した整数値が、正しい値かどうかを調べて、サーバ側のエンジンが書き換えられていないこと、つまり、正当なエンジンであることを検証する。

【００２２】クライアント側のエンジン上に、サーバ側エンジン４の正当性を検証するエージェントが動作している場合には、検証できた時点ではじめてモーバイルコードの発信を行うことによりセキュリティを確保する。

【００２３】またモーバイルコード自体の中に、サーバ側エンジン４の正当性を検証するエージェントのコードが有る場合には、本来の実行コードと持参データとを暗号化して、持参データとしておく。この暗号化は、正当なエンジンコードを一方向関数に入力したときの整数値を鍵として復号化できるようにしておく。モーバイルコ

ードが最初に、サーバ側エンジン４に到達したときには、サーバ側の検証代行エージェント９とメッセージ交換を行うことで、サーバ側のエンジンを一方向関数に入力したときの整数値を受け取る。その整数値を鍵として、暗号化してある本来の実行コードと持参データとを復号化して、本来のエージェントをサーバ側エンジン４上で動作させることで、セキュリティを確保する。

【００２４】次に図１の構成において、読み出しパターンと一方向関数を用いて行うエンジンの正当性検証処理の手法を、図２乃至図５により具体例で説明する。図２の（Ａ）は、複数バイトで表されるエンジンコードの例を示す。図２の（Ｂ）は、このようなエンジンコードに対して与えられる読み出しパターンの例を示す。読み出しパターンは、エンジンコードのバイト数に対応するビット数をもち、その順次のビット位置は、エンジンコードの順次のバイトデータに対応づけられている。従って、エンジンコードがバイト構成のメモリに格納された場合、読み出しパターンの各ビットは、メモリの順次の番地に対応することになる。

【００２５】読み出しパターンは、例えば乱数に基づいて生成される。読み出しパターンは、図２の（Ｂ）に示されるように、値“１”をもつビット位置に対応する番地のバイトデータのみを、メモリから読み出して、エンジンコードから、読み出しパターンに依存する少数のデータを抽出する。

【００２６】図３は一方向関数の機能を示す。図２で読み出しパターンにより読み出されたデータは、一方向関数に入力されて、入力データに依存する値の固定長の出力整数に変換される。

【００２７】図４に示されるように、正当性を検証しようとするサーバ側エンジンのエンジンコードは、クライアント側にも保持されている。クライアント側では、読み出しパターンを生成して、自己が保持するエンジンコードからデータを抽出し、一方向関数に入力して、クライアント側出力整数を作成する。一方、サーバ側で実行されるエンジンコードからも、同じ読み出しパターンを用いてデータを抽出させ、一方向関数によりサーバ側出力整数を作成させる。このようにして得られたクライアント側出力整数とサーバ側出力整数とを比較し、図示のように一致が得られた場合には、サーバ側エンジンは正当であると判定する。

【００２８】これに対し、図５に示されるように、不正処理の目的などでサーバ側エンジンコードの一部が書き替えられている場合には、図４と同じ処理を行っても、得られるクライアント側出力整数とサーバ側出力整数とは不一致となり、サーバ側エンジンコードは不正なものであると判定される。

【００２９】

【発明の実施の形態】図６は、本発明が適用可能なインターネット上のマルチエージェントシステムとモーバイルコードの実行を示す説明図である。図示のようにある計算機のエンジンから送信されたモーバイルコードは、インターネット上に分散している計算機のエンジン間を移動し、順次のサーバエージェントとメッセージ交換をしながら必要なデータを処理していく。

【００３０】図７は、クライアント側のエンジンで検証エージェントが動作するシステムの構成例を示す説明図である。図７では、図１に示したシステム中の構成要素と共通の要素は、同じ参照番号で示してある。図７において新たに示された要素は１４ないし１６であり、１４はクライアント側システムのエンジン、１５はエンジン１４上で動作する検証エージェント、１６はエンジン１４から送信されたモーバイルコードである。

【００３１】エンジンコードのすべてを読み出すのではなく、エンジンコード中で読み出しパターンで指定された番地のデータだけが読み出されるようになる。したがって、クライアント側のエンジン１４上で動作している検証エージェント１５からの検証要求時に、読み出しパターンを任意に指定することで、次のようにセキュリティのレベルを向上させることが可能となる。

【００３２】つまりサーバ側は、クライアント側からの検証要求毎にエンジンコードの読み出しと一方向関数の計算を行なう必要があるので、最初は正当なエンジンを用い、次からは不正エンジンを用いるということができなくなる。クライアントとサーバ間のエンジン検証のためのメッセージ交換の内容を、第三者が盗聴しても、同一の読み出しパターンが再び使われなければ、無意味なパターンと数字の対でしかない。また、副次的効果として、エンジンコードのすべてを読み出す必要がなくなるので、検証のための時間的コストを下げることが可能となる。

【００３３】この読み出しパターンを用いる方法には、検証をクライアント側のエンジン上で行う場合と、検証をサーバ側のエンジン上で行う場合の二つがあり、どちらの場合でも有効性は変わらない。

【００３４】図７の検証をクライアント側のエンジン上で行う場合の動作を、図８の制御フローを参照して説明する。図７のシステム構成では、クライアント側のエンジン１４上で、サーバ側のエンジンの正当性を検証する検証エージェント１５を動作させ、利用するサーバが動作するエンジン４に向けて、モーバイルコードを送信する前に、その相手先のエンジンの正当性を検証する。最初に、クライアント側の検証エージェント１５から、サーバ側の検証代行エージェント９にエンジン検証要求メッセージを送信する（１）。ここで、読み出しパターンを用いるときには、そのパターンも送信する。サーバ側の検証代行エージェント９は、エンジン４内のインターフェース部１０を介して、エンジン外のエンジンコード読み出し部１１及び一方向関数計算部１３でエンジンコードに依存した整数値を得て、その整数値をクライアン

ト側の検証エージェントに返している（２）。このシステム構成の利点は、相手先のエンジンの検証が終わるまで、モーバイルコード１６を送信（３）しなくても良いので、送信途中及び送信先での不正読み出しと書き換えの機会を減らせることである。また、検証エージェント作成にあたって，使用リソースに制限が少ないことも利点となる。

【００３５】このように、クライアント側のエンジン１４上で、サーバ側のエンジンの正当性を検証する検証エージェント１５を動作させて、サーバ側の検証代行エージェント９とエンジン検証のためのメッセージ交換を行なう場合には、それらのメッセージ交換に、相互の認証とメッセージの電子署名と暗号化を行なうようにしてもよい。そうすると、サーバ側の検証代行エージェントがクライアント側からの検証要求メッセージの真偽を確認してから、検証手続きを開始することで不要な情報の漏出を防ぐとともに、悪意により頻繁に偽の検証要求メッセージが送出された場合のサーバ側のエンジン過負荷状態を回避できる。また、クライアント側がサーバ側からの検証のための整数値のメッセージの真偽を確認することで、第三者の偽または改ざんされたメッセージ送信による検証の妨害を防ぐことが可能となる。

【００３６】次に、検証エージェントをサーバ側のエンジン上で動作させる場合について述べる。図９は、この場合にクライアント側から、利用するサーバが動作しているエンジンに対して、送信するモーバイルコードの例を示す。モーバイルコードは、実行コード部分と持参コード部分からなる。

【００３７】実行コード部分には、エンジンの正当性の検証と、検証後の暗号化されたコード等の復号化、そして、そのコードの実行を行なう、エンジン検証エージェントのコードが格納される。また持参データ部分には、暗号化された本来の実行コードとその持参データとが格納されている。その暗号化は、エンジンコードを一方向関数に入力して得られる整数値を鍵として復号化できるように施されている。モーバイルコードが、サーバ側のエンジンに到達すると、エンジン検証エージェントが起動される。エンジン検証エージェントは、図１０の制御フローに示されるように、最初に、検証代行エージェントに対して、検証要求メッセージを送信する。ここで，読み出しパターンを用いるときには、そのパターンも送信する。検証代行エージェント９は、エンジン内のインターフェース部１０を介して、エンジン外のエンジンコード読み出し部、一方向関数計算部でエンジンコードに依存した整数値を得て、その整数値を検証エージェントに返す。検証エージェントは、その整数値を鍵として、持参データ中のこのエンジン上で起動すべきエージェントの実行コードと持参データの復号化を行い、このエンジン上で起動を行なう。このシステム構成の利点は、検証のためのメッセージ交換が、サーバ側のエンジン上でローカルに行なわれるので、第三者に不正な読み出しや書き換えを行なわれる可能性が低いことである。

【００３８】このように、検証エージェントをサーバ側のエンジン上で動作させる場合で、クライアント側から、利用するサーバが動作しているエンジンに対して、送信するモーバイルコードの持参データとして、本来の実行コードとそのデータとを暗号化して格納してあって、それらを検証代行エージェントから得たエンジンコードを一方向関数に入力して得られる整数値を鍵として、復号化して実行するシステムの場合とは、サーバ毎に固有の鍵を暗号化のときに付加する。暗号化に公開鍵を用いる場合で、サーバから得たデータを新たに持参データにする場合には、図１１に示すようにクライアントまたは途中でそのデータを利用する必要のあるサーバの公開鍵暗号を用いた暗号化を行なう。この方式の利点は、モーバイルコードが移動しながら複数のサーバを利用する場合に、途中の所定のサーバ及びクライアント自身以外に、実行コード及びデータの読み出し・書き換えを行なわれる危険性をさらに減らせることである。

【００３９】さらに、図１２に示すように、サーバ側にエンジンの自己検証装置１７を付加する構成とすることができる。この場合、自己検証装置１７は、サーバのエンジンの起動及び、稼動中に、定期的にエンジン外のエンジンコード読み出し部１１と一方向関数計算部１３でエンジンコードに依存した整数値を得て、正しい値と比較を行なうことにより、エンジンコードの書き換えの自己検証を行なう。この方式の利点は、サーバ側で自己検証を行なうことで、クライアント側のモーバイルコードの不正な読み出し・書き換えを未然に防ぐことで、サーバの信頼を保つことができることである。

【００４０】

【発明の効果】本発明によれば、モーバイルコードが実行される際に、モーバイルコードが持つデータ（実行コードを含む）の内、サーバに開示されるのは、モーバイルコード内の明示的記述により指定された部分のみであることを、サーバに用いられているモーバイルコード実行エンジン自体が保証し、その保証の確認をクライアント側でることができる。したがって、モーバイルコードの発信に際して、発信先のサーバに対しては、そのサーバにおける動作に関してモーバイルコード中の必要な情報のみを相手サーバに開示することが可能となるので、インターネット上の複数のサーバを巡っての情報収集や複数ステップからなる一連のデータ処理作業のように、個々のサーバにはモーバイルコード中の一部分の情報を開示しても良いが全部を開示したくない場合や、モーバイルコードが各々のサーバから得て内蔵している情報を他のサーバには秘密にする必要がある場合に、モーバイルコードを安全に用いることが可能となる。

【図面の簡単な説明】

【図１】本発明の原理的構成図である。

11

【図２】本発明によるエンジンの正当性の検証処理（その１）の説明図である。

【図３】本発明によるエンジンの正当性の検証処理（その２）の説明図である。

【図４】本発明によるエンジンの正当性の検証処理（その３）の説明図である。

【図５】本発明によるエンジンの正当性の検証処理（その４）の説明図である。

【図６】インターネット上でのマルチエージェントシステムとモーバイルコードの実行を示す説明図である。

【図７】クライアント側のエンジンで検証エージェントが動作するシステムの構成例を示す説明図である。

【図８】クライアント側のエンジン上で検証エージェントを動作させるシステムの制御フローである。

【図９】サーバ側のエンジン上で検証エージェントを動作させるシステムのモーバイルコードの説明図である。

【図１０】サーバ側のエンジン上で検証エンジンを動作させるシステムの制御フローを示す図である。

12

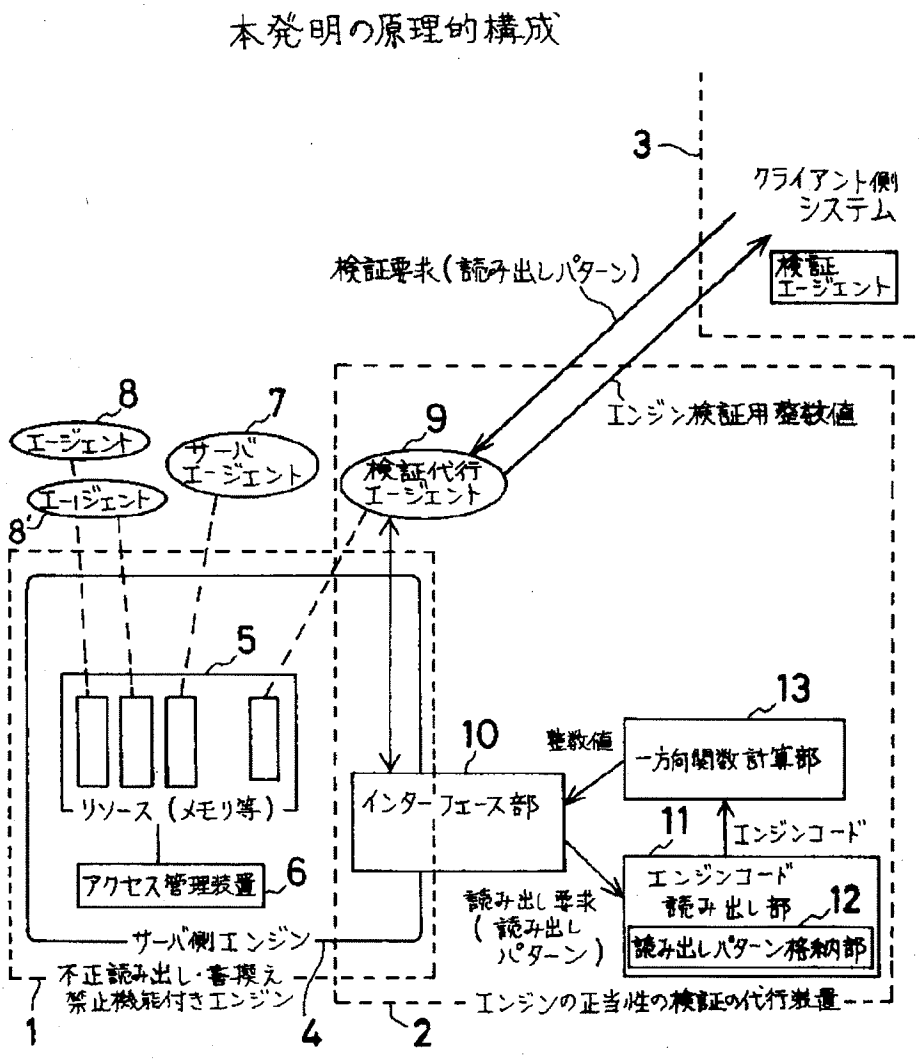
＊【図１１】モーバイルコードの構成例を示す説明図である。

【図１２】エンジンに自己検証装置を付加した構成例を示す説明図である。

【符号の説明】

１：　不正読み出し・書換え禁止機能付きエンジン
２：　エンジンの正当性の検証の代行装置
３：　クライアント側システム
４：　サーバ側エンジン
５：　リソース
６：　アクセス管理装置
７：　サーバエージェント
８：　エージェント
９：　検証代行エージェント
１０：インターフェース部
１１：エンジンコード読み出し部
１２：読み出しパターン格納部
１３：一方向関数計算部

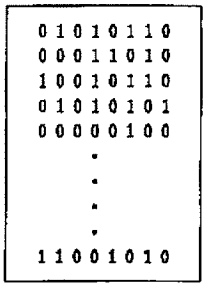
【図１】



【図２】

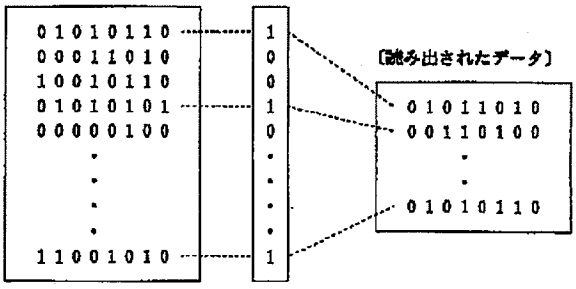
本発明によるエンジンの正当性の検証処理（その１）

（A）エンジンコードの例



（B）読み出しパターンによる読み出しの例

〔エンジンコード〕　〔読み出しパターン〕　〔読み出されたデータ〕



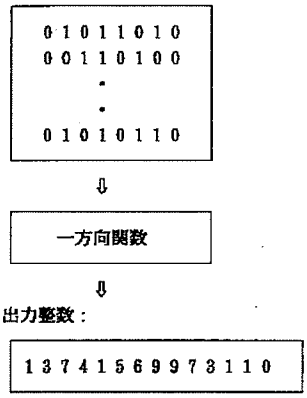
【図３】

本発明によるエンジンの正当性の検証処理（その２）

〔一方向関数の機能〕

読み出しパターンにより

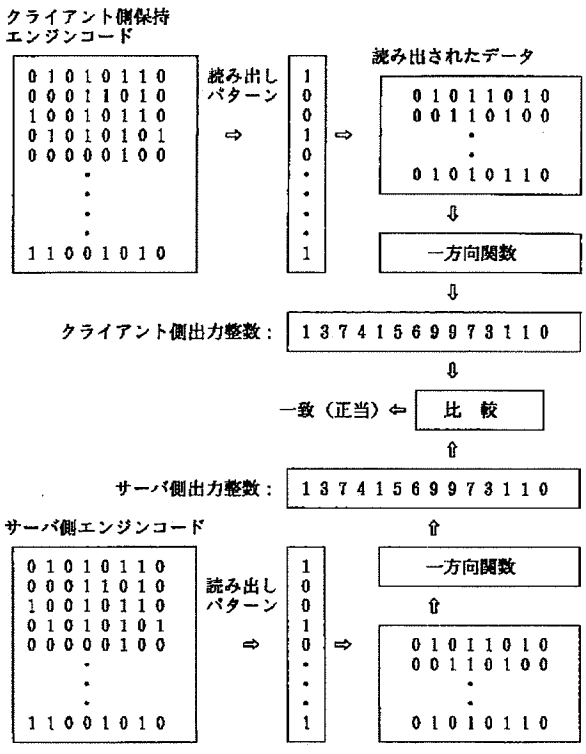
読み出されたデータ：



【図４】

本発明によるエンジンの正当性の検証処理（その３）

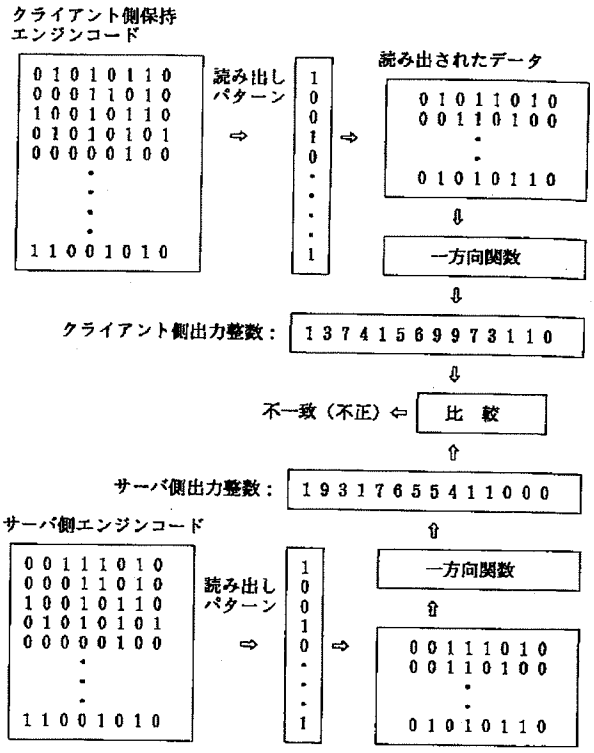
〔エンジンコードが正当な場合〕



【図５】

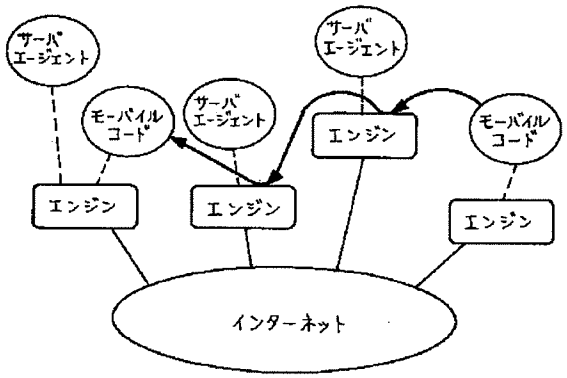
本発明によるエンジンの正当性の検証処理（その４）

〔エンジンコードが不正な場合〕



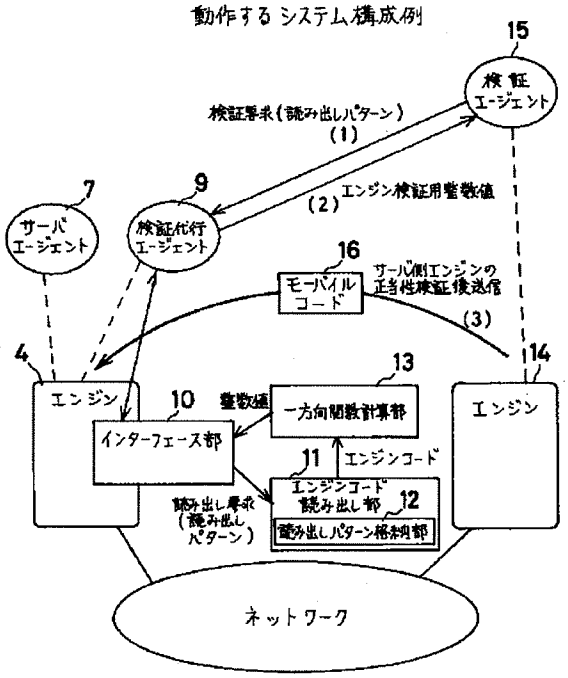
【図６】

インターネット上でのマルチエージェントシステムと
モーバイルコードの一例



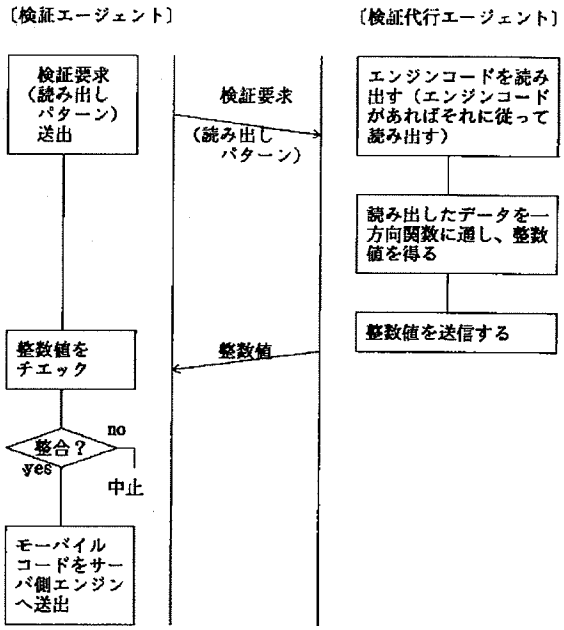
【図７】

クライアント側のエンジンで検証エージェントが
動作するシステム構成例



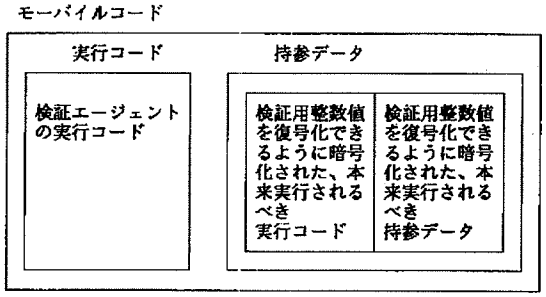
【図８】

クライアント側のエンジン上で検証エージェントを動作
させるシステムの制御フロー



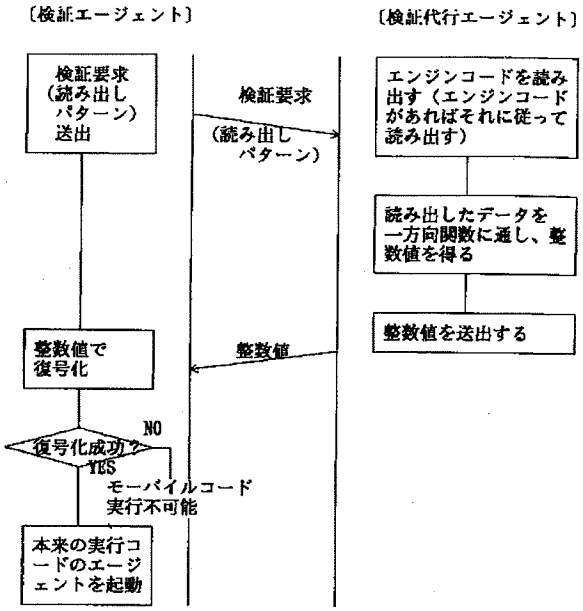
【図９】

サーバ側のエンジン上で検証エージェントを動作させるシステム
のモーバイルコード



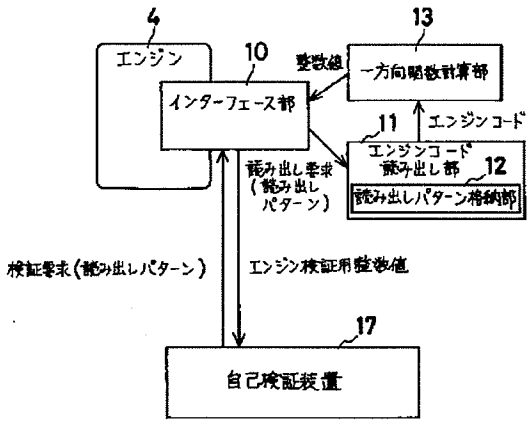
【図１０】

サーバ側のエンジン上で検証エージェントを動作させるシステムの制御フロー



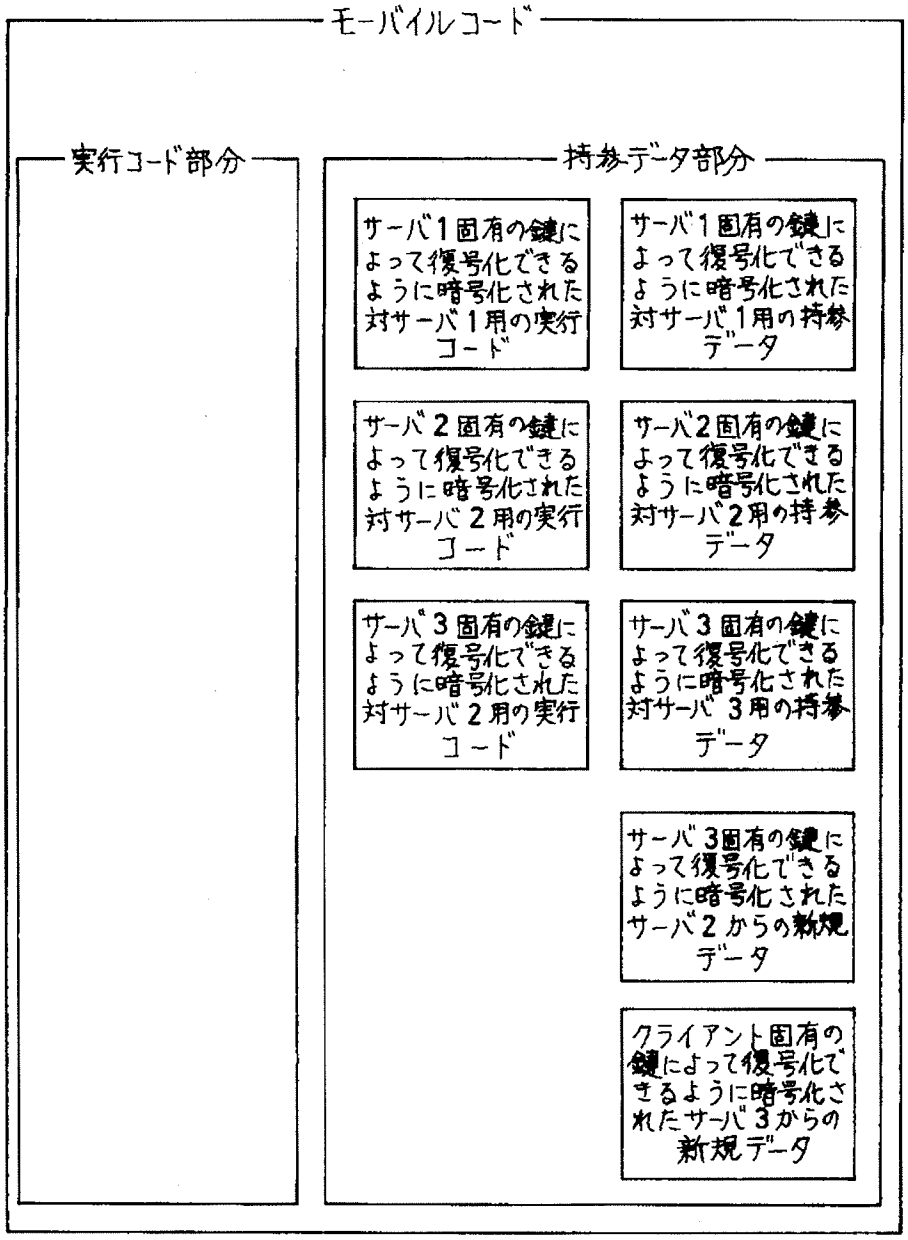
【図１２】

エンジンに自己検証装置を付加した構成例



【図１１】

モーバイルコードの構成例



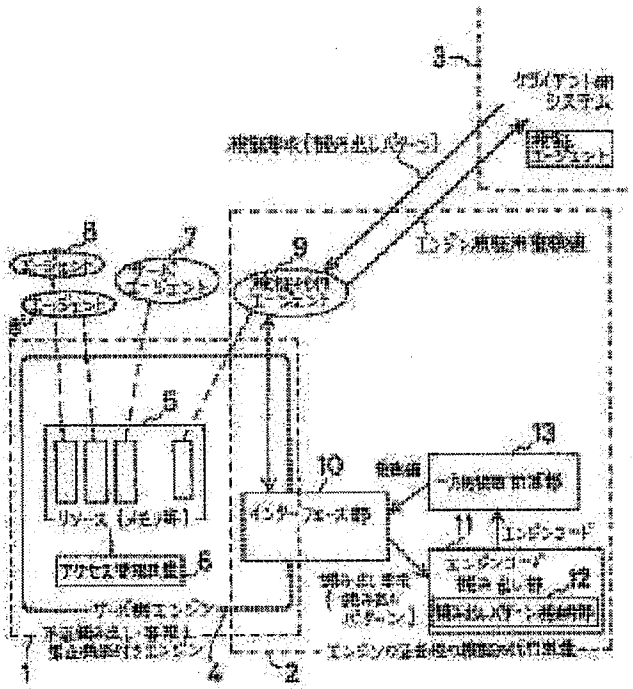
MOBILE CODE EXECUTION SYSTEM

Publication number: JP10254840 (A)
Publication date: 1998-09-25
Inventor(s): KAWAMURA AKIRA
Applicant(s): FUJITSU LTD
Classification:
- international: ***G06F15/16; G06F9/06; G06F9/46; G06F9/50; G06F9/54; G06F13/00; G06F15/00; G06F21/20; G06F21/22;*** **G06F15/16; G06F9/06; G06F9/46; G06F13/00; G06F15/00; G06F21/20; G06F21/22;** (IPC1-7): G06F15/16; G06F9/06; G06F13/00; G06F15/00
- European:
Application number: JP19970058651 19970313
Priority number(s): JP19970058651 19970313

Also published as:
JP3905170 (B2)

Abstract of **JP 10254840 (A)**

PROBLEM TO BE SOLVED: To ensure that a mobile code is not processed illicitly in executing the mobile code by checking an integer value obtained by inputting the code of an engine to one direction function and verifying that the engine code is not rewritten. SOLUTION: A verification substitute agent 9 which substitutively verifies the justness of the engine 4 operates on the server-side engine 4. The verification substitute agent 9 obtains the integer value depending on the engine code by an engine code read part 11 and a one direction function calculation part 13, which are out of the engine, through an interface part 10 in the engine 4 in accordance with the request message of a client-side verification agent and returns the integer value to the verification agent of the client.; When the engine code is rewritten for realizing the illicit processing, the integer value which one direction function outputs differs from that of the original engine code. Thus, rewriting can be detected.



* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1]A mode BAIRU code execution system comprising:
Execution engine with a mechanism in which read-out and rewriting of information without necessity in a MOBA yl code are forbidden when executing a MOBA yl code.
A device which verifies that an engine code is not rewritten by investigating an integral value acquired by inputting a code of the engine into a one-way function.

[Claim 2]In the MOBA yl code execution system according to claim 1, add read-out pattern storage to an engine code read section, and all the engine codes are not read to it, A MOBA yl code execution system reading only data of an address specified by a read-out pattern.
[Claim 3]In the MOBA yl code execution system according to claim 1, on an engine of a client side, A MOBA yl code execution system characterized by verifying the justification of an engine of the partner point before transmitting a MOBA yl code towards an engine with which an agent who verifies the justification of an engine by the side of a server is operated, and the server to be used operates.
[Claim 4]A MOBA yl code execution system characterized by performing a mutual verification, an electronic signature of a message, and encryption in the MOBA yl code execution system according to claim 3 at the time of message switching for engine verification by the side of a server.
[Claim 5]In the MOBA yl code execution system according to claim 1, about composition contents of a MOBA yl code which transmits, as an execution code, As data which decryption of a code after verification of engine justification and verification etc. which were enciphered, and a code of an agent for engine verification of the server side which executes a decrypted code were stored, and was brought, A MOBA yl code execution system, wherein an execution code and data of enciphered original are stored, and the encryption is given so that an integral value acquired by inputting an engine code by the side of a server into a one-way function can be decrypted as a key.
[Claim 6]In the MOBA yl code execution system according to claim 5, about encryption with an original execution code and its data. In adding for every server at the time of encryption of a peculiar key and using a public key for encryption, A MOBA yl code execution system performing encryption using public key encryption of a server which needs to use the data in the middle of a client when newly using as bringing data data obtained from a server.
[Claim 7]An engine self-verification device is added in the MOBA yl code execution system according to claim 1, It is acquiring periodically an integral value for which it depended on an engine code by an engine code read section and a one-way function calculation part besides an engine during the time of starting of an engine of a server, and operation, and performing comparison with a right value, A MOBA yl code execution system performing self-verification of rewriting of an engine code.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention]In information processing with the multi-agent system according especially to a MOBA yl cord about the security art of the information processing in the environment where the computer of a large number which exist this invention dispersedly was connected in the network, It is related with the MOBA yl cord executive system it is guaranteed that does not perform unjust processing to the MOBA yl code at the time of execution of a MOBA yl cord.
[0002]By development of a personal computer, and development of computer network art. Make a mainframe into a main host, a terminal hangs down from it, and each personal computer workstation is made into a front end from the centralized processing method which depends for all the information processing on a main host, The client server method of choosing suitably the database server using a computer with the calculation server and the large-scale memory storage using two or more supercomputers installed on the network, and processing information in the purpose is becoming in use.
[0003]The frame of the used computer or communication media was overcome by the open standard as technology trends of a worldwide computer network, and the Internet which is the network-of-network formed by connection of each computer network is expanded quickly. The motion is expanded also from the field of not only the field of a scale but use from a specialist to the general public, and the Internet exceeds the existing frameworks, such as a company and a country, and it is continuing growing as a computer network with worldwide breadth. The intranet which limited the use area to the specific in-house is also spreading these days, inheriting the pliability and convenience of the Internet.
[0004]Along with the expansion in the scale and field-of-the-invention both sides of such Internet intranet, to all the things which exist on a network from what of the range to which the relation between client servers was also restricted is [being exclusive and] fixed. To sometimes respond and to evolve into what can be connected and used freely came to be desired. Therefore, an interface which is not bound to an available plat form is needed. To provide advanced service in which the user who uses unifies and treats the server of the kind from which two or more networks differ with a simple interface as he spreads from the limited specialist to the general public came to be needed. In order to fill those requests, the multi-agent system in which the agent of the side to access realizes advanced information processing by exchange of a message with the agent who manages each server attracts attention.
[0005]The correspondence to the case where it has not always connected with a network by the spread of personal digital assistants, In order to send in an agent not from message switching with an agent ready-made [by the side of a server] but from here and to perform advanced and flexible information processing, The MOBA yl cord (MOBA yl agent) method that the agent itself moves in a network top is thought as important as a part of multi-agent system.
[0006]
[Description of the Prior Art]A multi-agent's system is implemented as follows. That is, the engines (shell, an interpreter, etc.) which manage at least one, generation and disappearance of

an agent, and communication between agents are placed on each computer. And substance, such as a process and a thread, is given so that each agent may function in mutually-independent and may exchange messages mutually. Each agent's substance may be a process with independent case where it exists in an engine process and engine. The message switching with the agent who exists on another computer on a network, It is indirectly carried out directly among agents by considering an engine as agency via an engine using TCP (data link control protocol) or UDP (User Datagram Protocol).

[0007]In a MOBA yl cord method, the cord of the agent who should perform is transmitted as a message from one computer, and the cord is performed by the received computer. Therefore, implementation which generally carries out immediate execution of the sent cord on the system of a computer is not taken, but implementation used as the virtual machine for performing the interpreter or byte cord for performing the cord for an engine is taken. It does so because [of security reservation of the abstraction for considering it as a multiplatform, and the system of the computer described later].

[0008]In the Internet which is the environment which the computer is distributing on a network, and a network which has worldwide breadth especially, As the pliability and convenience being inside-out, who accesses what by what kind of means from where when, it succeeds in what, and there is possibility (danger) about the influence of the result coming out how infinitely. Therefore, it is necessary to restrict access by bad faith or inattention. For the realization, installing the fire wall which used the computer for access controls in the entrance of the network of the regular range, and adding restriction to the network of the partner point which communicates, and the kind of communications protocol is performed on the network level.

[0009]In the level of a multi-agent's engine. In order to prevent adding restriction to the range, quantity, and occupation time of the resource which can be accessed for restriction of the open range, and impossible prevention of the system by superfluous occupancy of operation, and unlawful access to systems, such as computer virus, Always carrying out indirect access to the system of a computer, or performing code verification before execution of a MOBA yl cord etc. is performed.

[0010]There is no telling along what kind of course the Internet generally passes on the structure, but there are tapping and an alteration of the communication data on a course, and a possibility that a holder in bad faith will impersonate an original partner and danger. In order to prevent it, art, such as encryption of a communication content and an electronic signature for guaranteeing attestation and not being altered of the partner adapting encoding technology, is used. Specifically, PEM (Privacy-Enhanced Mail), PGP (Pretty Good Privacy), etc. using Kerberos using a secret key cryptosystem, or a public key and a secret key cryptosystem exist.

[0011]

[Problem(s) to be Solved by the Invention]The protection as an object of intellectual-property-rights reservation is needed as MOBA yl cord itself has advanced features. Usage which performs a drawer, processing, etc. of data one after another may also be performed by sending a MOBA yl cord to a network in the form accompanied by data, and carrying out two or more service servers and message switching on a network. In that case, since the service server on a network cannot always trust it, the necessity of keeping secret the data which needed to prevent read-out of unnecessary data to data processing, and was obtained from a certain service server from other service servers is also possible.

[0012]However, the conventional security art aims at protecting data and the system in a network from external unlawful access etc., and the protection about the MOBA yl cord which transmits is not enough. Although the server which provides service was kept as one of the agents and there was also an engine prevented from peeping into the contents of the MOBA yl cord unjustly from a server, a means to verify whether the engine really has the limiting function did not exist.

[0013]

[Means for Solving the Problem]In a MOBA yl cord executive system of this invention. When a system by the side of a server performs a MOBA yl cord, it comprises an engine with a function which forbids unjust read-out and rewriting of an execution code portion in a MOBA yl cord, and

a bringing data part, and a device which executes verification with the just engine by proxy. Verification of whether an engine function to forbid unjust read-out rewriting of data in a MOBA yl code is just is performed by checking a code of the engine.

[0014]An engine has in an inside a controlling device which performs an access control of resources, such as a memory used while each agent performs, so that other agents cannot access a field which each agent has not permitted clearly.

[0015]Drawing 1 is a principle lineblock diagram of this invention. In drawing 1, 1 is a vicarious execution device of verification of the justification of unjust read-out and an engine with a rewriting prohibition function by this invention, and an engine according [2] to this invention. 3 is a client side system which requires service. 4 is server side engine in which unjust read-out and the engine 1 with a rewriting prohibition function are realized. 5 is resources, such as a memory which it has in the server side engine 4. An access management device with which 6 manages access to the resource 5, and 7 thru/or 8 is the agents who are operating using the resource 5, and server agent, 8, and 8' is other agents 7. 9 thru/or 13 constitute the vicarious execution device 2 of verification of engine justification, and 9 is a verification vicarious execution agent who operates on the server side engine 4. 10 is an interface part in an engine. 11 is an engine cord read section which exists on the primary storage besides an engine, or a secondary storage, Read-out pattern storage in which a pattern which specifies an address of data which 12 reads from an engine cord is stored, and 13 are one-way function calculation parts which consider data of an engine cord read from the engine cord read section 11 as an input, carry out functional calculus, and output a fixed-length integral value.

[0016]The client side system 3 has the following two kinds of composition. The agent to whom one of them verifies the justification of an engine by the side of a server operates on an engine of a client side. As for other one, an original execution code and bringing data which were enciphered are in the MOBA yl cord itself, and there is a cord of an agent for performing decryption and execution of a cord of verification of engine justification, an enciphered cord after verification, etc. further. In the case of which [the], information on a cord of an engine by the side of a server is stored at a client side.

[0017]A MOBA yl code system transmits an execution code and bringing data which should be protected from unjust read-out and rewriting to a computer by which the server to be used is operating. Therefore, reservation of security of a MOBA yl cord must be performed in an engine with which a MOBA yl cord is performed. However, generally there is no server side engine 4 under management of the client side system 3. Therefore, in this invention, unjust read-out and rewriting as the server side engine 4 using what is made impossible from a client side. By making it verifiable whether an engine by the side of a server is just, security of a MOBA yl cord sent out from a client side is secured.

[0018]The server side engine 4 has in an inside the access management device 6 which performs an access control of the resources 5, such as a memory used while each agent performs, so that other agents cannot access a field which each agent has not permitted clearly. And a server which provides service implements service as one of the agents. If it does so, it will become impossible from each service of a server to write other than data which a MOBA yl cord specified clearly.

[0019]On the server side engine 4, the verification vicarious execution agent 9 who executes verification of the justification of the engine 4 by proxy is operating. The verification vicarious execution agent 9 via the interface part 10 in the engine 4 according to a request message of a verification agent of a client side, An integral value for which it depended on an engine cord by the engine cord read section 11 and the one-way function calculation part 13 besides an engine is acquired, and the integral value is returned to a verification agent of a client. Here, a one-way function is also called a secure hash function and a Messge Digest function, and, specifically, are MD4, MD5, SHA, checksum CRC, etc. A one-way function has the following character.

[0020]1. considering arbitrary sequences of bytes as an input -- a fixed-length integral value -- output 2., if it rewrites a difficult engine cord for unjust processing realization that discovery of an input which gives the same output defines the difficulty 3. inverse function, Since an integral value which a one-way function outputs differs from the time of the original engine cord

rewriting is detectable. Although it is possible to become the value same by chance naturally, if a size of an integral value becomes enough, it will be stored in probability which is satisfactory practically. Since discovery of an input which gives the same output is difficult, it is difficult to perform additional rewriting for consistency doubling for escaping detection of rewriting to a rewritten engine cord. Here, what the server side provides a client side with is only an infinite meaningless integral value as it is, in view of a viewpoint of security by the side of a server. The integral value has a meaning only after a client side knows the contents of the engine cord. That is, for the server side, it is offer of the minimum information, and it will be said that verification is possible.

[0021]A verification agent of a client side investigates whether an integral value corresponding to the engine cord is a right value, and verifies that an engine by the side of a server is not rewritten, i.e., it is a just engine.

[0022]Security is secured by sending a MOBA yl cord for the first time on an engine of a client side, when the agent who verifies the justification of the server side engine 4 is operating, and it is able to verify.

[0023]When a cord of an agent who verifies the justification of the server side engine 4 is in the MOBA yl cord itself, an original execution code and bringing data are enciphered and it is considered as bringing data. This encryption enables it to decrypt an integral value when a just engine cord is inputted into a one-way function as a key. First, when a MOBA yl cord reaches the server side engine 4, it receives an integral value when an engine by the side of a server is inputted into a one-way function by performing the verification vicarious execution agent 9 and message switching by the side of a server. Security is secured in decrypting an original execution code and bringing data which have been enciphered by using the integral value as a key, and operating an original agent on the server side engine 4.

[0024]Next, in composition of drawing 1, an example explains the technique of engine justification verification processing performed using a read-out pattern and a one-way function by drawing 2 thru/or drawing 5. (A) of drawing 2 shows an example of an engine code expressed with two or more bytes. (B) of drawing 2 shows an example of a read-out pattern given to such an engine code. A read-out pattern has the number of bits corresponding to a number of bytes of an engine code, and the sequential bit position is matched with sequential byte data of an engine code. Therefore, when an engine code is stored in a memory of byte composition, each bit of a read-out pattern will correspond to a sequential address of a memory.

[0025]A read-out pattern is generated, for example based on a random number. As shown in (B) of drawing 2, a read-out pattern reads only byte data of an address corresponding to a bit position with a value "1" from a memory, and extracts a small number of data depending on a read-out pattern from an engine code.

[0026]Drawing 3 shows a function of a one-way function. Data which read by drawing 2 and was read with a pattern is inputted into a one-way function, and is changed into a fixed-length output integer of a value depending on input data.

[0027]As shown in drawing 4, an engine code of server side engine which is going to verify justification is held also at a client side. In a client side, a read-out pattern is generated, data is extracted from an engine code which self holds, it inputs into a one-way function, and a client side output integer is created. On the other hand, data is made to extract also from an engine code executed by the server side using the same read-out pattern, and the server side output integer is made to create by a one-way function. Thus, an acquired client side output integer is compared with the server side output integer, and when coincidence is obtained like a graphic display, it judges with server side engine being just.

[0028]On the other hand, as shown in drawing 5, when a part of server-side-engine code is rewritten for the purpose of unjust processing, etc. Even if it performs the same processing as drawing 4, it becomes inharmonious [a client side output integer acquired and the server side output integer], and is judged with a server-side-engine code being inaccurate.

[0029]

[Embodiment of the Invention]Drawing 6 is an explanatory view showing the multi agent system on the Internet which can apply this invention, and execution of a MOBA yl code. The MOBA yl

code transmitted from the engine of a certain computer like a graphic display moves between the engines of the computer currently distributed on the Internet, and it processes required data, carrying out a sequential server agent and message switching.

[0030]Drawing 7 is an explanatory view showing the example of composition of the system by which the verification agent operates with the engine of a client side. In drawing 7, the same reference number has shown the component in the system shown in drawing 1, and the common element. The elements newly shown in drawing 7 are 14 thru/or 16, and the verification agent to whom 14 operates with the engine of a client side system, and 15 operates on the engine 14, and 16 are the MOBA yl codes transmitted from the engine 14.

[0031]Only the data of the address which did not read all the engine codes, but is among an engine code, read, and was specified by the pattern comes to be read. Therefore, it becomes possible to raise the level of security as follows by specifying a read-out pattern as the verification demand from the verification agent 15 who is operating on the engine 14 of a client side arbitrarily.

[0032]Since the server side needs to perform read-out of an engine code and calculation of a one-way function for every verification demand from a client side, it becomes impossible that is, to say that an inaccurate engine is used from the next using a just engine at first. If the same read-out pattern is not again used even if a third party intercepts the contents of the message switching for the engine verification between a client and a server, it is only a pair of a meaningless pattern and a number. Since it becomes unnecessary to read all the engine codes as a secondary effect, it becomes possible to lower the time cost for verification.

[0033]There are two, the case where it verifies on the engine of a client side, and when verification is performed on the engine by the side of a server, in the method of using this read-out pattern, and even when it is which, validity does not change.

[0034]The operation in the case of verifying drawing 7 on the engine of a client side is explained with reference to the flows of control of drawing 8. In the system configuration of drawing 7, the verification agent 15 who verifies the justification of the engine by the side of a server is operated on the engine 14 of a client side, and before transmitting a MOBA yl code towards the engine 4 with which the server to be used operates, the justification of the engine of the partner point is verified. First, an engine verification request message is transmitted to the verification vicarious execution agent 9 of the server side from the verification agent 15 of a client side (1). Here, when using a read-out pattern, the pattern also transmits. The verification vicarious execution agent 9 of the server side via the interface part 10 in the engine 4, The integral value for which it depended on the engine code by the engine code read section 11 and the one-way function calculation part 13 besides an engine was acquired, and the integral value is returned to the verification agent of the client side (2). Since the advantage of this system configuration does not need to transmit the MOBA yl code 16 until verification of the engine of the partner point finishes (3), it is being able to reduce unjust read-out by the transmission destination in the middle of transmitting, and the opportunity of rewriting. It becomes an advantage in verification agent creation that the resource used also has little restriction.

[0035]Thus, the verification agent 15 who verifies the justification of the engine by the side of a server is operated on the engine 14 of a client side, When performing the verification vicarious execution agent 9 of the server side, and message switching for engine verification, it may be made to perform mutual attestation, the electronic signature of a message, and encryption to those message switching. While preventing a break through of unnecessary information by starting verification procedure after the verification vicarious execution agent of the server side checks the truth of the verification request message from a client side if it does so, The engine overloaded state by the side of a server when a fake verification request message is frequently sent out by bad faith is avoidable. It becomes possible to prevent disturbance of verification by a third party's imitation or altered message transmission by checking the truth of the message of the integral value for the verification from the server side of a client side.

[0036]Next, the case where a verification agent is operated on the engine by the side of a server is described. Drawing 9 shows the example of the MOBA yl code which transmits from a client side to the engine with which the server to be used is operating in this case. A MOBA yl code

consists of an execution code portion and a bringing code part.
[0037]The code of the engine verification agent who performs verification of engine justification, decryption of the code after verification etc. which were enciphered, and execution of the code is stored in an execution code portion. The enciphered original execution code and its bringing data are stored in the bringing data part. The encryption is given so that the integral value acquired by inputting an engine code into a one-way function can be decrypted as a key. If a MOBA yl code reaches the engine by the side of a server, an engine verification agent will be started. An engine verification agent transmits a verification request message to a verification vicarious execution agent first, as shown in the flows of control of drawing 10. Here, when using a read-out pattern, the pattern also transmits. The verification vicarious execution agent 9 acquires the integral value for which it depended on the engine code by the engine code read section besides an engine, and the one-way function calculation part via the interface part 10 in an engine, and returns the integral value to a verification agent. A verification agent performs decryption of an agent's execution code and bringing data which should be started on this engine in bringing data by using that integral value as a key, and starts on this engine. Since message switching for verification is locally performed on the engine by the side of a server, the advantage of this system configuration is that a possibility that unjust read-out and rewriting will be performed for a third party is low.
[0038]By thus, the case where a verification agent is operated on the engine by the side of a server. As opposed to the engine with which the server to be used is operating from the client side, As bringing data of a MOBA yl code to transmit, encipher an original execution code and its data, have stored, and the integral value acquired by inputting into a one-way function the engine code which obtained them from the verification vicarious execution agent is used as a key, With the case of the system decrypted and performed, it adds for every server at the time of encryption of a peculiar key. In newly using as bringing data the data obtained from the server by the case where a public key is used for encryption, it performs encryption using the public key encryption of the server which needs to use the data in the middle of a client as shown in drawing 11. The advantage of this method is being able to reduce further the danger read-out and rewriting of an execution code and data being performed in addition to an intermediate predetermined server and the client itself, when using two or more servers, while a MOBA yl code moves.
[0039]As shown in drawing 12, it can have composition which adds the engine self-verification device 17 to the server side. In this case, the self-verification device 17 during starting of the engine of a server, and operation, Self-verification of rewriting of an engine code is performed by acquiring the integral value for which it depended on the engine code periodically by the engine code read section 11 and the one-way function calculation part 13 besides an engine, and comparing with a right value. the advantage of this method being performing self-verification by the server side, being preventing unjust read-out and rewriting of the MOBA yl code of a client side, and maintaining reliance of a server — ****** — they are things.
[0040]
[Effect of the Invention]Being indicated by the server among the data (an execution code is included) which a MOBA yl code has, when a MOBA yl code is executed according to this invention, The MOBA yl code execution engine itself used [that it is only the portion specified by the explicit description in a MOBA yl code and] for the server can guarantee, and the check of the guarantee can be carried out by a client side. Therefore, the server of a calling destination is received on the occasion of dispatch of a MOBA yl code, Since it becomes possible to disclose only the required information in a MOBA yl code to a partner server about the operation in the server, Like a series of data processing operation which consists of the information gathering and plural steps involving two or more servers on the Internet, Although a part of information in a MOBA yl code may be disclosed to each server, when you do not want to indicate all, or when the information which a MOBA yl code acquires from each server, and builds in needs to be made secret at other servers, it becomes possible to use a MOBA yl code safely.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

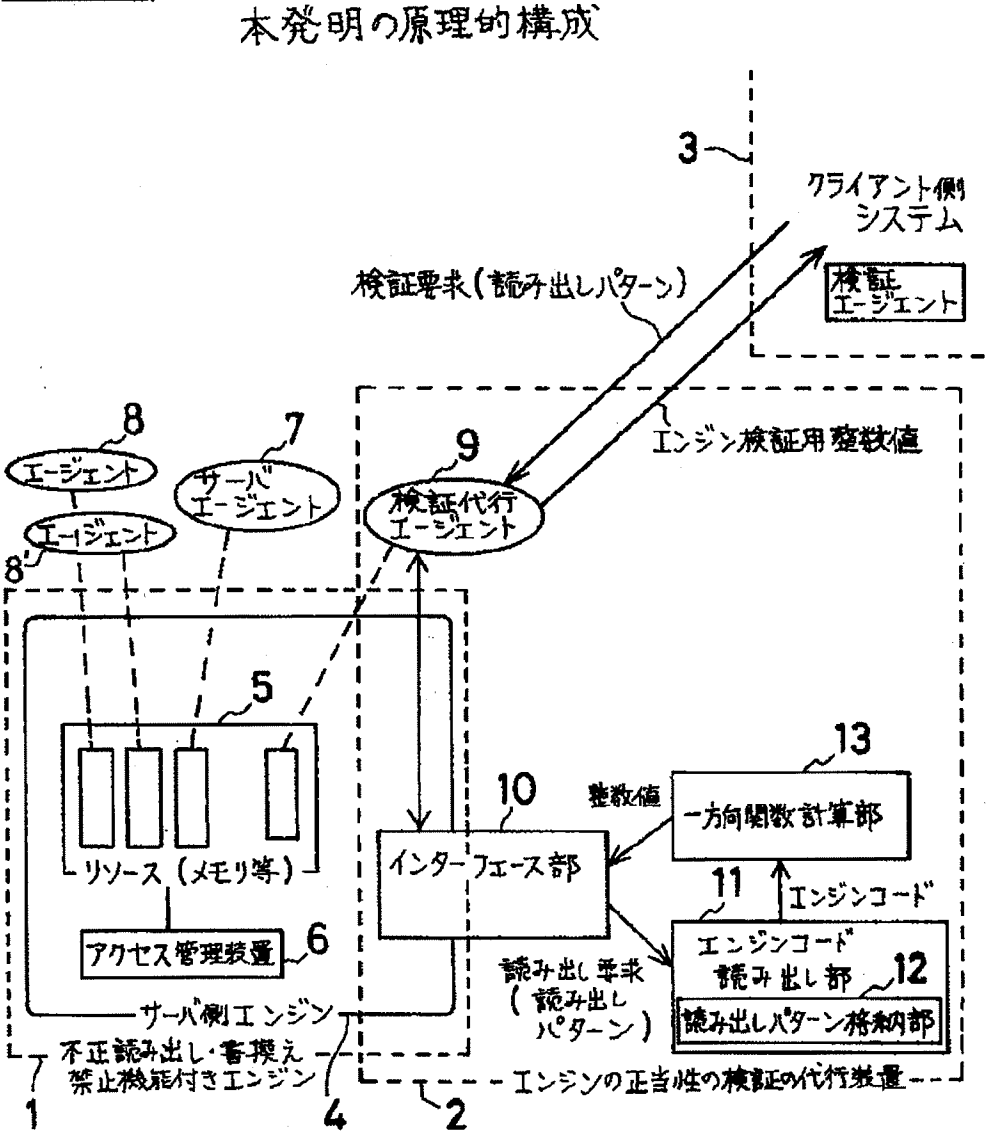
DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a theoretic lineblock diagram of this invention.
[Drawing 2]It is an explanatory view of the verification processing (the 1) of the justification of the engine by this invention.
[Drawing 3]It is an explanatory view of the verification processing (the 2) of the justification of the engine by this invention.
[Drawing 4]It is an explanatory view of the verification processing (the 3) of the justification of the engine by this invention.
[Drawing 5]It is an explanatory view of the verification processing (the 4) of the justification of the engine by this invention.
[Drawing 6]It is an explanatory view showing the multi agent system on the Internet, and execution of a MOBA yl code.
[Drawing 7]It is an explanatory view showing the example of composition of the system by which the verification agent operates with the engine of a client side.
[Drawing 8]They are flows of control of a system which operate a verification agent on the engine of a client side.
[Drawing 9]It is an explanatory view of the MOBA yl code of the system which operates a verification agent on the engine by the side of a server.
[Drawing 10]It is a figure showing the flows of control of a system which operate a verification engine on the engine by the side of a server.
[Drawing 11]It is an explanatory view showing the example of composition of a MOBA yl code.
[Drawing 12]It is an explanatory view showing the example of composition which added the self-verification device in an engine.
[Description of Notations]
1: Unjust read-out and an engine with a rewriting prohibition function
2: A vicarious execution device of verification of engine justification
3: Client side system
4: Server side engine
5: Resource
6: Access management device
7: Server agent
8: Agent
9: Verification vicarious execution agent
10: Interface part
11: Engine code read section
12: Read-out pattern storage
13: One-way function calculation part

[Translation done.]

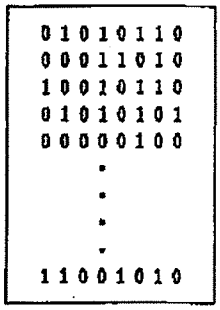
DRAWINGS

[Drawing 1]

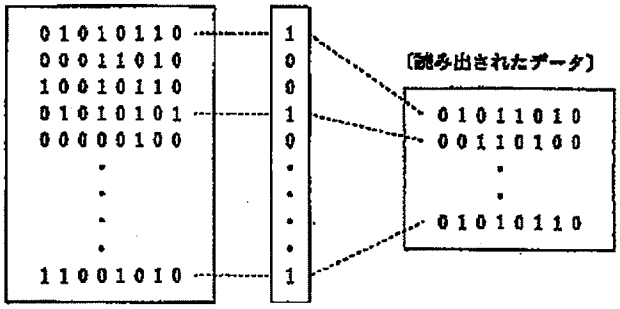


[Drawing 2]

本発明によるエンジンの正当性の検証処理（その１）
（A）エンジンコードの例

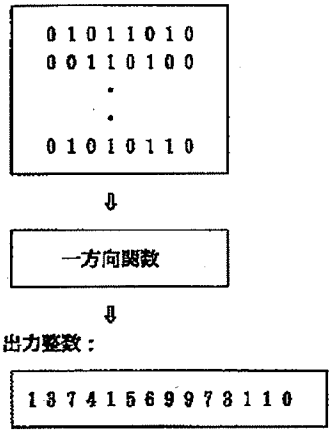


（B）読み出しパターンによる読み出しの例



[Drawing 3]

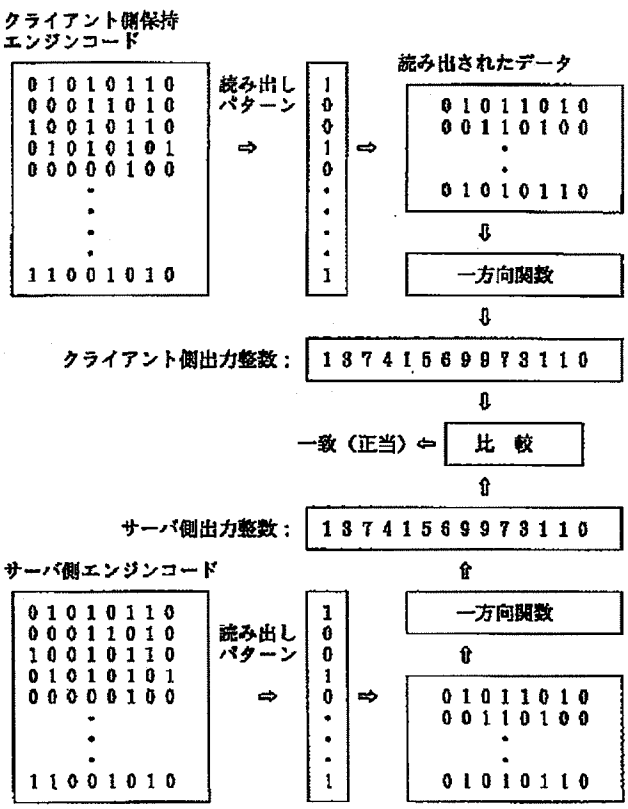
本発明によるエンジンの正当性の検証処理（その２）
〔一方向関数の機能〕
読み出しパターンにより
読み出されたデータ：



出力整数：

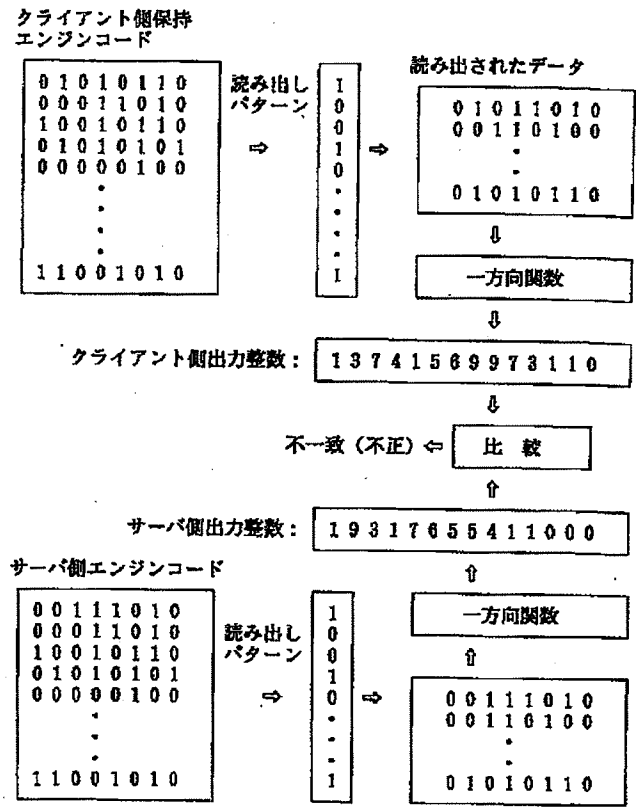
[Drawing 4]

本発明によるエンジンの正当性の検証処理（その３）
〔エンジンコードが正当な場合〕



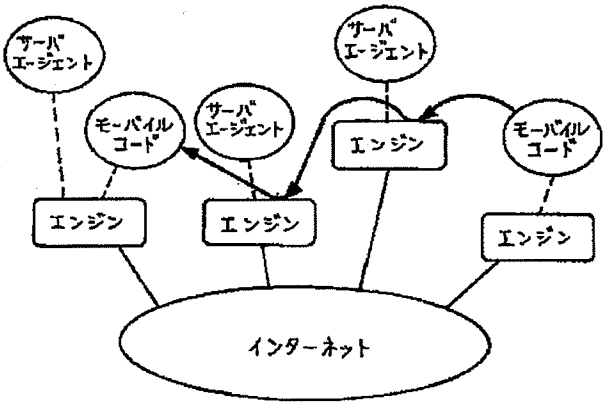
[Drawing 5]

本発明によるエンジンの正当性の検証処理（その４）
〔エンジンコードが不正な場合〕



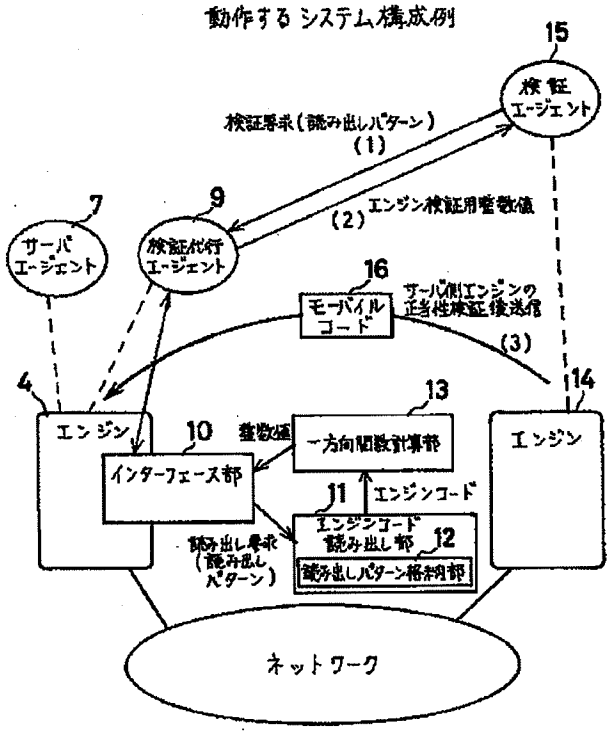
[Drawing 6]

インターネット上でのマルチエージェントシステムと
モーバイルコードの一例



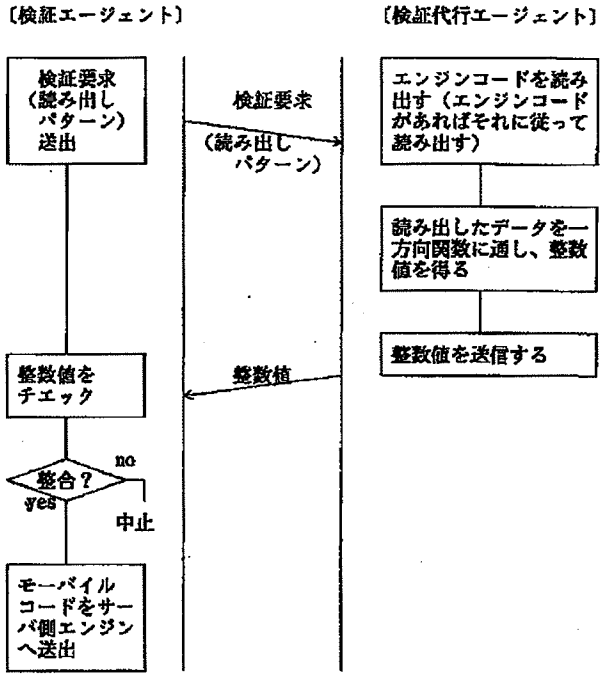
[Drawing 7]

クライアント側のエンジンで検証エージェントが
動作するシステム構成例



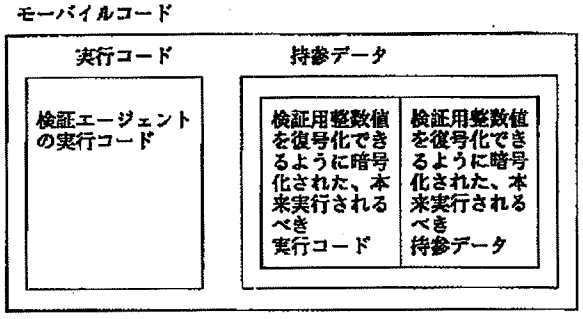
[Drawing 8]

クライアント側のエンジン上で検証エージェントを動作させるシステムの制御フロー



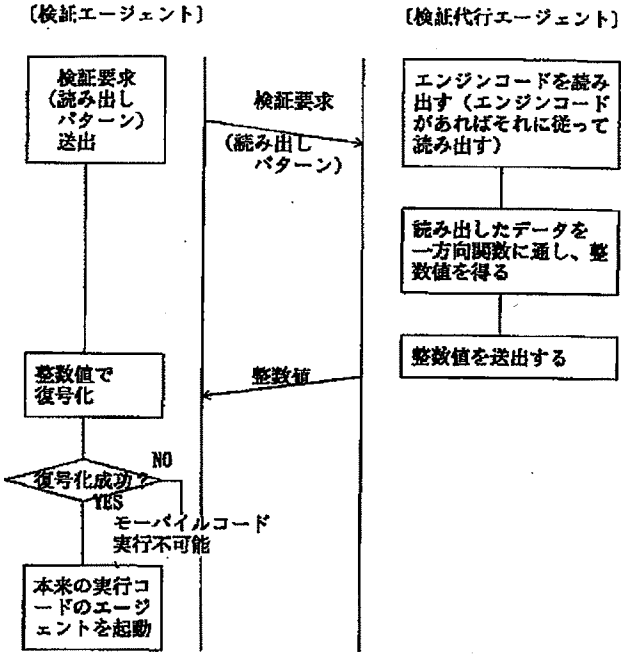
[Drawing 9]

サーバ側のエンジン上で検証エージェントを動作させるシステムのモーバイルコード



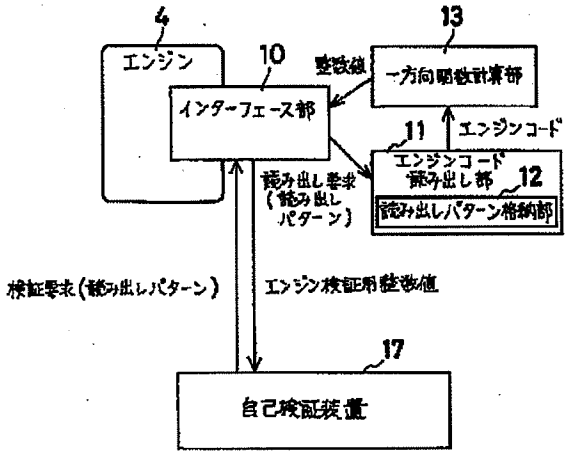
[Drawing 10]

サーバ側のエンジン上で検証エージェントを動作させるシステムの制御フロー



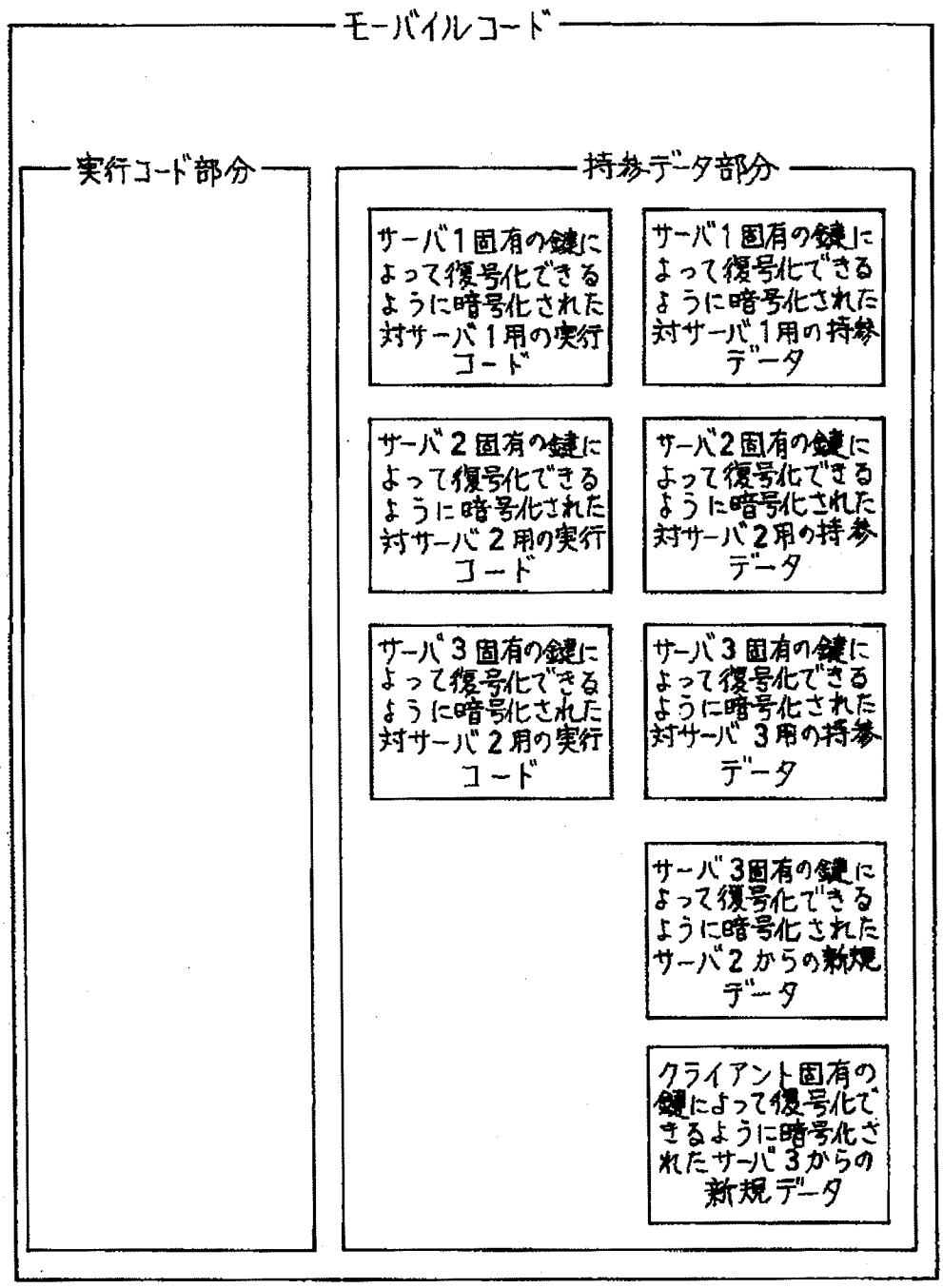
[Drawing 12]

エンジンに自己検証装置を付加した構成例



[Drawing 11]

モーバイルコードの構成例



[Translation done.]